

הנחיות כלליות

- המתרגל האחראי על תרגיל זה הוא: **אייל**.
 - יש לשלוח את הקבצים באמצעות **מערכת ההגשה** לפני חלוף התאריך **11/01/22**.
 - ניתן להגיש את התרגיל באיחור עם קנס אוטומטי על פי הפירוט הבא:
 - 👁️ יום איחור (**12/01/22**) - קנס של **10 נקודות** (ציון מקסימלי ~~90~~).
 - 👁️ יומיים איחור (**13/01/22**) - קנס של **20 נקודות** (ציון מקסימלי ~~80~~).
 - 💣 לאחר מכן לא יהיה ניתן להגיש את התרגיל (ציון 0).
 - שאלות בנוגע לתרגיל יש לפרסם **באופן ציבורי בפורום הקורס** בלבד!
 - בקשות להארכה (מסיבות מוצדקות בלבד, כגון: מילואים, לידה וכו') יש לפרסם **באופן פרטי בפורום הקורס** בלבד (יש למען את הפוסט ל-instructors).
 - בכל בקשה יש לציין:
 - 👤 שם מלא.
 - 💻 שם משתמש במערכת ההגשה.
 - 💾 תעודת זהות.
 - יש להקפיד מאוד על הוראות עיצוב הקלט והפלט, בדיוק על פי הדוגמאות המצורפות.
 - בנוסף שימו לב להנחיות במסמך ה-Coding Style המפורסם באתר הקורס.
- עליכם לכתוב קוד על פי ההנחיות ולוודא שקיבלתם 100 בבדיקה האוטומטית הראשונית, וכן שהתרגיל מתקמפל ורץ על השרת המחלקתי (planet) ללא **שגיאות** או **אזהרות**.
- תרגיל שלא עומד בסטנדרטים הבסיסיים הללו יגרור **ירידה משמעותית בציון התרגיל**, בשל הטרחה שהוא מייצר בתהליך הבדיקה שלו, עד כדי ציון 0.
- להזכירכם העבודה היא אישית. "עבודה משותפת" דינה כהעתקה. התרגיל נבדק על ידי מערכת ההגשה האוטומטית גם מהבחינה הזו, ותרגיל שהועתק יגרור ציון 0 **לכל הגורמים** השותפים בהעתקה. אתם יכולים לדון בגישות לפתרון התרגיל באופן תיאורטי, אך אין לשתף קוד בשום צורה.
- בפיתוח הקוד ניתן להשתמש בכל סביבת עבודה, העיקר הוא שתדעו איך לקחת את קבצי הקוד מתוך הסביבה הזו, לבדוק אותם על שרתי האוניברסיטה ולהגיש אותם באמצעות מערכת ההגשה.
- דוגמאות לחלק מהסביבות האפשריות:
- Visual studio
 - Clion
 - Atom
 - Sublime
 - Notepad++
- שימו לב שאתם מגישים אך ורק את הקבצים המכילים את הקוד שלכם, ולא קבצים מיותרים שנוצרו על ידי סביבת העבודה. כמו כן הימנעו מהגשת קבצים/תיקיות עם שמות המכילים תווים בעברית.
- מומלץ להגיש קובץ ריק כדי לקבל במייל את הפלט המדויק שמערכת ההגשה מצפה לראות (מבחינת מחרוזות והודעות למשתמש), ולהתאים את ההדפסות בתוכנית לפלט הזה.

בתרגיל זה עליכם ליצור ולהגיש תוכנית המחולקת לקבצים הבאים (אך יש להגיש את הקבצים המודגשים בלבד):

ex_6.c **person.c** person.h **sorting.c** sorting.h

בתרגיל זה מותר להשתמש בספריות string.h stdlib.h stdio.h בתוך הקובץ person.c בלבד. ניתן להשתמש בכל החומר שנלמד בתרגולים. אין להשתמש בפונקציה qsort מתוך הספרייה stdlib.

פקודת הקימפול בתרגיל זה היא:

gcc ex_6.c person.c sorting.c -lm -std=c99 -DNDEBUG

או בקיצור

gcc *.c -lm -std=c99 -DNDEBUG

כל הקוד והקבצים המתוארים במסמך זה ניתנים להורדה מהפיאצה, אין צורך להעתיק טקסט מהמסמך.

המודול sorting

בקובץ ההצהרות (H) של המודול הזה מופיעה החתימה הבאה:

```
// including start, excluding stop
```

```
void quicksort(void* arr, int start, int stop, void(*swap)(void*, int, int), int(*compare)(void*, int, int));
```

בתוך הקובץ sorting.zip/qs (שנמצא בפיאצה) יש שני מימושים למיון מהיר (הראשון בפונקציה quicksort והשני בפונקציה qsort). אחד המימושים תקין, והשני כולל באג. עליכם לקחת מימוש תקין של המיון (או על ידי בחירה במימוש התקין או על ידי בחירה במימוש עם הבאג לאחר שתיקנתם אותו), ולהתאים את האלגוריתם לחתימה הגנרית כך שהפונקציה תקבל פונקציית החלפה ופונקציית השוואה.

בקובץ sorting.c תופיע הפקודה הבאה:

```
#include "sorting.h"
```

ולאחר מכן המימוש שלכם למיון הגנרי.

שימו לב שהמיון כולל את הטווח שמתחיל באינדקס start ומסתיים באינדקס stop-1 (כולל) - וזו גם הגישה בשני המימושים שבקובץ sorting.zip.

אם בחרתם לתקן את הבאג, צרפו להגשה שלכם קובץ **טקסט פשוט** נוסף בשם bug.txt, המכיל תיאור קצר באנגלית המסביר מה היה הבאג ואיך תיקנתם אותו. היזהרו לא להגיש קובץ עם שם עברי ולא לכתוב הסבר בעברית, משום שזה יכול לפגוע בהרצה האוטומטית של התרגיל שלכם וכך ציונכם ייפגע.

המודול person

בקובץ ההצהרות (H) של המודול הזה מופיע הקוד הבא:

```
#define MAX_STR_LEN 100
#define DELIM ","
#define DELIM_CHAR ','

typedef struct {
    int day;
    int month;
    int year;
} Date;

typedef struct {
    char firstName[MAX_STR_LEN];
    char lastName[MAX_STR_LEN];
    Date dateOfBirth;
    int height;
    double weight;
} Person;

int comparePersonByFirstName(void* arr, int i, int j);
int comparePersonByLastName(void* arr, int i, int j);
int comparePersonByWeight(void* arr, int i, int j);
int comparePersonByHeight(void* arr, int i, int j);
int comparePersonByBMI(void* arr, int i, int j);
int comparePersonByDate(void* arr, int i, int j);
void swapPersons(void* arr, int i, int j);
void load(const char* inputFile, Person persons[], int* numOfPersonsPtr);
void save(const char* outputFile, Person persons[], int numOfPersons);
```

לצורך נוחות ופישוט של התרגיל, אנחנו חורגים מעט מהאופן האידיאלי שבו יש לכתוב את התוכנית, מהבחינות הבאות:

- הגדרת מבנים בקובץ H.

- הגבלת גדלים בעזרת define במקום להשתמש בהקצאה דינאמית.

הטיפוס "איש" מוגדר להיות שם פרטי, שם משפחה, תאריך לידה (יום, חודש, שנה), גובה ומשקל.

קידוד של "איש" כמחרוזת טקסט יתבצע כך שכל השדות ייכתבו ברצף עם DELIM ביניהם, לדוגמה:

ברברה לופז, שנולדה בתאריך 4.2.1964, הגובה שלה הוא 191 סנטימטרים ומשקלה 70.51 ק"ג תקודד כמחרוזת טקסט באופן הבא:

Barbara,Lopez,4,2,1964,191,70.51

ניתן להניח שכל מחרוזת כזו המופיעה בקובץ הקלט לא תהיה ארוכה יותר מגודלו של MAX_STR_LEN.

בקובץ person.c עליכם לממש:

- פונקציות השוואה לפי כל אחד מהשדות האלו, ובנוסף פונקציית השוואה לפי ערך ה-BMI.
- השוואת שמות תתבצע לפי סדר מילוני (לקסיקוגרפי) רגיל.
- השוואת תאריכים מוגדרת לפי סדר הזמנים - כלומר תאריך קטן התרחש לפני תאריך גדול (הבהרה כי נראה שזה לא היה ניסוח ברור: לאחר מיון אדם מבוגר יופיע לפני אדם צעיר).
- מדד BMI מוגדר להיות המשקל חלקי הגובה בריבוע (וניתן להסתדר בלי math.h לצורך החישוב הזה).
- פונקציית החלפה.
- פונקציית טעינה של מערך אנשים מתוך קובץ טקסט.
- פונקציית שמירה של מערך אנשים לתוך קובץ טקסט.

ניתן (ומומלץ) להגדיר פונקציות עזר ולעשות בהן שימוש ככל שתמצאו, אך שימו לב שיש להגדיר אותן כסטטיות, כדי שלא לפגוע בשאר הרכיבים בתוכנית.

עבודה עם קבצים היא נושא טכני שהחלטנו שאין צורך להקדיש לו שיעור, ולכן הוא נשאר לימוד עצמי. מצורפת כאן פונקציה פשוטה שקוראת קובץ קלט וכותבת אותו (שורה שורה) אל קובץ פלט, ניתן להתבסס עליה בעבודה על התרגיל.

```
#include <stdio.h>
void ioExample(const char* inputFile, const char* outputFile) {
    FILE* input = fopen(inputFile, "r");
    if (!input) {
        printf("File not found: %s\n", inputFile);
        return;
    }
    FILE* output = fopen(outputFile, "a");
    if (!output) {
        printf("Error with file: %s\n", outputFile);
        return;
    }
    char buffer[MAX_STR_LEN];
    fprintf(output, "-----\n");
    while (fgets(buffer, MAX_STR_LEN, input)) {
        fprintf(output, "%s", buffer);
    }
    if (fclose(input)) {
        printf("Error with closing file: %s\n", inputFile);
        return;
    }
    if (fclose(output)) {
        printf("Error with closing file: %s\n", outputFile);
    }
}
```

שימו לב שמכיוון שקובץ הפלט נפתח במוד "a", אם נריץ את הפונקציה כמה פעמים היא תשרשר את הפלט לקובץ במקום לדרוס אותו (מוזמנים להריץ ולנסות).

את הקובץ ex_6.c אין צורך להגיש, הוא יכלול את פונקציית ה-main ויראה בסגנון הזה:

```
#include "sorting.h"
#include "person.h"
#define MAX_LINES 100
const char* INPUTFILE = "persons.txt";
const char* OUTPUTFILE = "sorted.txt";
int main() {
    Person persons[MAX_LINES];
    int numOfPersons = 0;
    load(INPUTFILE, persons, &numOfPersons);
    quicksort(persons, 0, numOfPersons, swapPersons, comparePersonByFirstName);
    save(OUTPUTFILE, persons, numOfPersons);
    return 0;
}
```

בנוסף הוא מדפיס את הקבצים לפני ואחרי המיון כדי שתוכלו לראות את הפלט במייל.

כאן נעשית הנחה שבקובץ הקלט לא יהיו יותר שורות מאשר MAX_LINES.

התוכנית לא מדפיסה דבר אל המסך, אך לאחר ההרצה עם הקובץ persons.txt אמור להיווצר הקובץ sorted.txt כפי שהם מופיעים בדוגמה הבאה (בהתאם לפונקציית ההשוואה שנבחרה):

<u>persons.txt</u>	<u>sorted.txt</u>
James,Smith,12,10,2005,152,110.05	Barbara,Lopez,8,4,1989,191,181.21
Mary,Johnson,10,10,1956,156,93.53	Charles,Jackson,14,3,1955,195,95.41
Robert,Williams,5,2,1983,161,161.84	David,Hernandez,17,4,1951,151,122.90
Patricia,Brown,23,5,1971,156,77.84	Elizabeth,Martinez,11,11,2000,160,86.83
John,Jones,2,3,1957,195,129.09	James,Smith,12,10,2005,152,110.05
Jennifer,Garcia,27,4,2001,197,157.33	Jennifer,Garcia,27,4,2001,197,157.33
Michael,Miller,13,10,1952,153,116.87	Jessica,Thomas,15,11,1951,179,123.76
Linda,Davis,1,10,1954,191,86.54	John,Jones,2,3,1957,195,129.09
William,Rodriguez,4,8,1985,157,87.41	Joseph,Anderson,3,1,1967,177,88.58
Elizabeth,Martinez,11,11,2000,160,86.83	Karen,Martin,28,4,1959,167,190.67
David,Hernandez,17,4,1951,151,122.9	Linda,Davis,1,10,1954,191,86.54
Barbara,Lopez,8,4,1989,191,181.21	Mary,Johnson,10,10,1956,156,93.53
Richard,Gonzalez,4,7,2005,156,50.9	Michael,Miller,13,10,1952,153,116.87
Susan,Wilson,17,4,1960,193,84.07	Patricia,Brown,23,5,1971,156,77.84
Joseph,Anderson,3,1,1967,177,88.58	Richard,Gonzalez,4,7,2005,156,50.90
Jessica,Thomas,15,11,1951,179,123.76	Robert,Williams,5,2,1983,161,161.84
Thomas,Taylor,9,8,1991,184,176.77	Sarah,Moore,2,6,1996,183,149.29
Sarah,Moore,2,6,1996,183,149.29	Susan,Wilson,17,4,1960,193,84.07
Charles,Jackson,14,3,1955,195,95.41	Thomas,Taylor,9,8,1991,184,176.77
Karen,Martin,28,4,1959,167,190.67	William,Rodriguez,4,8,1985,157,87.41

שימו לב שקבצים בסביבת חלונות (windows) נבדלים מקבצים בסביבת יוניקס/לינוקס בתו ירידת השורה שלהם (בחלונות - \r\n ביוניקס/לינוקס - \n). לפעמים קבצים שעוברים בין המערכות לא מוצגים טוב בעורכי טקסט שונים. ניתן להשתמש בפקודות dos2unix,unix2dos ולאחריהן שם הקובץ על מנת לבצע המרה לפורמט הרצוי. בסיבת הבדיקה של התרגיל הקבצים יהיו כמובן בסגנון יוניקס/לינוקס, אך מומלץ לכתוב את הקוד כך שהוא יעבוד בכל הסביבות.

שאלות נפוצות

- האם ניתן להניח שקובץ הקלט תקין, ושהמידע יופיע בהתאם לשדות של person?
- כן. לא יהיה מידע נוסף או חסר - בדיוק כמו בקובץ שמופיע בדוגמה.
- ניתן להניח ששם הקובץ שמועבר לפונקציה save מייצג קובץ שאינו קיים, ולכן זה לא משנה אם תשתמשו במוד "w" או במוד "a".
- בקובץ הפלט יש לשמור את המספרים העשרוניים עם דיוק של שתי ספרות אחרי הנקודה.
- יש להתייחס לכישלון בפתיחת/סגירת קבצים כמו שהתייחסתי בקוד לדוגמה. נוסח ההדפסה לא קריטי, זה יבדק ידנית ולא אוטומטית.
- האם צריך להמיר את הגובה למטרים לפני חישוב ערך ה-BMI?
- זו הגדרת הנוסחה, לכן בהחלט עדיף לעבוד על פי ההגדרה (ואולי זה לא משנה, אבל לא בדקתי).
- תיקנתי את הקובץ על השרת כך שעכשיו הוא לא מכיל ערכים זהים (וכך יהיה גם בבדיקה).